



## Matriz de Soluciones

**Laboratorio TUP - 2021**

Comisión 2

Profesora: Carolina Archuby

**Integrantes:**

Fazio, Galo

Rodriguez, Lucía

Seijo, Lucas

## Funciones para ABMCL de Usuarios y Películas.

### **Alta, Baja y Modificación de Usuarios.**

Estas funciones se encuentran en el apartado "Usuario" (existe otro llamado Películas), el cual está disponible tanto para los administradores como para los usuarios, con distintos privilegios según el caso.

Sus funciones principales son *"altaBajaDeUsuario"*, *"modificarUsuariosEnArchivoAdmin"* y *"modificarUsuariosEnArchivo"*, de allí se desprende el uso de otras funciones para lograr los objetivos de cada una.

```
printf("\n      ::: USUARIO :::\n");
printf("\n* 1 - Alta de usuario");
printf("\n* 2 - Baja de usuario");
printf("\n* 3 - Modificacion de datos");
printf("\n* 4 - Mostrar datos");
printf("\n* 5 - Volver\n");
```

#### *Alta de usuarios*

Para el usuario, se presentará el mensaje *"Su usuario ya ha sido dado de alta"* en caso de que ingrese la opción 1.

Al administrador le dará la opción de dar de alta a un usuario que ya está en el sistema (por ejemplo, en caso de que el mismo se arrepienta y decida volver a utilizarlo), o ingresar a un usuario nuevo.

#### *Baja de usuarios*

Si el usuario ingresa la opción 2, le preguntará si está seguro de que quiere darse de baja. En caso de hacerlo, este usuario pasará a tener el estado de "inactivo" (visible por el *admin*), pero no se lo eliminará de la lista.

En caso de que el administrador ingrese la opción 2, deberá ingresar el ID del usuario que desee dar de baja del sistema.

#### *Modificación de usuarios*

Cuando el usuario ingresa la opción 3, se le presenta la opción de modificar de su propia cuenta el nombre, mail, celular o contraseña. Cuando lo hace el administrador, se le pide ingresar el ID del usuario a modificar, y este mismo puede cambiar cualquier dato del mismo, incluyendo su estado en el sistema (como si llamara a la función de Alta o Baja, pero en este caso cambiará el estado mediante un string que indique el estado: *inactivo/activo*).

## Alta, Baja, Consulta y Listados de Películas.

Para este apartado, los menús presentados según el caso son distintos.

```
printf("\n      :: MENU PELICULAS :: \n");
printf("\n* 1 - Listado de peliculas");
printf("\n* 2 - Agregar peliculas a mi lista");
printf("\n* 3 - Quitar peliculas de mi lista");
printf("\n* 4 - Recomendaciones para vos");
printf("\n* 5 - Tu lista");
printf("\n* 6 - Buscar una pelicula");
printf("\n* 7 - Volver\n");
```

Menú presentado al usuario.

```
printf("\n      :: MENU PELICULAS :: \n");
printf("\n* 1 - Listado de peliculas");
printf("\n* 2 - Alta de peliculas");
printf("\n* 3 - Baja de peliculas");
printf("\n* 4 - Modificacion de datos de pelicula");
printf("\n* 5 - Consultar por una pelicula");
printf("\n* 6 - Volver\n");
```

Menú presentado al administrador.

### Listado de películas

Tanto al usuario como al administrador, al ingresar la opción 1, se le presentarán por pantalla 3 opciones: listar por orden alfabético, año o género las películas. Todas las películas del archivo se pasarán a un arreglo, y, según la opción que se ingrese, dicho arreglo se ordenará alfabéticamente teniendo en cuenta el nombre de las películas mediante la función *“mostrarListadoPorNombre”*, o por año de la más nueva a la más vieja con *“mostrarListadoPorAntiguedad”*: sea cual sea el criterio por el que se ordenen las películas en el arreglo, se utilizará el ordenamiento por selección.

También pueden filtrarse películas según el género que el usuario o administrador ingrese a un string *genero[]* en la función *“filtroGenero”*.

### Alta de películas

Al ingresar la opción 2, el administrador podrá agregar, mediante *“altaBajaDePelicula”*, una película al sistema con todos sus datos, como ser el nombre, año, género, 3 actores/actrices, calificación y estado de la misma.

En cambio, el usuario deberá ingresar el nombre de una película para añadirlo al arreglo *“miLista”*, individual para cada usuario y modificable por el mismo.

### Baja de películas

Cuando el administrador ingrese la opción 3, podrá dar de baja del sistema las películas que desee a través de la función *“altaBajaDePelicula”*, dejando de mostrárselas al usuario pero manteniéndolas en el archivo.

Para el usuario, ingresar esta opción significará quitar una película de su lista personal.

### Consulta de películas

La opción 6 para el usuario realiza la búsqueda de una película en el archivo del sistema mediante la función *“buscarMostrarPeliculaPorNombre”*, la cual asignará el valor “1” a un flag en caso de que la comparación de strings entre el nombre de cada película del archivo y el nombre de la película que le llegó por parámetro sean iguales. Caso contrario, imprimirá por pantalla *“No se encontró la película”*.

El administrador en cambio, debe ingresar 5 para ejecutar esta función: podrá hacer una consulta ingresando el nombre de una película y el sistema le imprimirá por pantalla sus respectivos datos en caso de que exista.

### Recomendación de películas

La opción 4 para el usuario le devolverá un listado con películas del género que más se repite en su lista.

Para el cometido de recomendar películas a un usuario se crearon 2 nuevas funciones y se utilizaron, en total, 4 funciones en el main.

En este caso, *“recomendacionesPorGeneroFavorito”* busca al usuario que está intentando recibir recomendaciones en el archivo, y al encontrarlo se detiene en su arreglo *“miLista”* para comparar cada posición del mismo con los strings correspondientes al nombre de cada género (Accion, Comedia, Drama, Romantica, Suspenso, Terror).

Con cada *strcmp* que realiza, a la vez cuenta cuántas veces se encuentra cada género en la lista de favoritos del usuario, y por último, llama a la función *“generoMasVisto”*, pasándole por parámetro la cantidad de películas de cada género en la lista del usuario y el arreglo de caracteres *“generoFavorito”* que fue declarado en el *main*.

La función *“generoMasVisto”* toma cada cantidad y la compara con las demás. El género que se identifique como favorito va a ser el que se copie en el string *“generoFavorito”*. Luego, en el *main*, se asigna a *“validosFavoritas”* el valor que retorne *“filtroGenero”*, que recibe el archivo de películas, un arreglo de tipo Película y un string que le indica el género que debe filtrar. Se obtiene de esta manera la cantidad de posiciones realmente cargadas del *“arregloFavoritas”*.

Por último, se llama a la función *“mostrarArreglo”*, pasándole por parámetro el arreglo de películas favoritas y sus válidos, para que le muestre al usuario las películas disponibles del género que más se repite en su lista *“miLista”*.

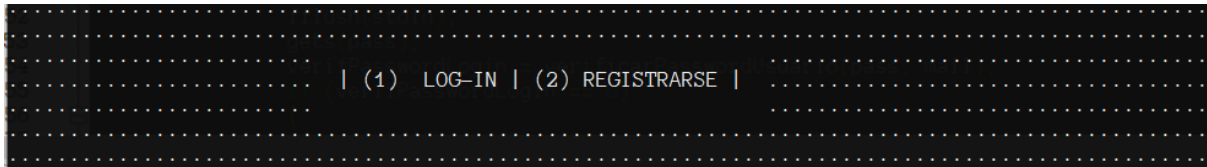
#### *Mostrar listado personal de películas*

La opción de mostrar un listado personal es exclusiva para el usuario. Ingresar la opción 5 del apartado “Películas” le permitirá, al usuario que se haya logueado, ver su arreglo de películas favoritas impreso en pantalla como un listado. Toda película que se haya ingresado al *array* mediante *“cargarMiListaUsuario”* será mostrado con la función *“mostrarMiLista”*.

## Funciones para manejo de archivos binarios

### **Persistencia de datos en el archivos - Validación de usuarios y películas**

Al principio del código, luego de la bienvenida del sistema, se presentan dos opciones:



Si se ingresa la opción 1, con la función *“verificarMailUsuario”* se comprobará mediante la variable *“verifMailLogin”* (declarada en el main e inicializada en 0) si el mail efectivamente se encuentra registrado en el sistema. El flag que retorne la función será determinante: si es 1, se mostrará por pantalla *“El mail ingresado no se encuentra registrado”*, pidiéndole al usuario que vuelva a ingresar una dirección e-mail. En cambio, si es 0, procederá a pedirle que ingrese su contraseña.

Aquí se hará presente la función *“verificarPasswordUsuario”*, que va a verificar si la contraseña que se ingresa es correcta y si concuerda con el mail que se escribió anteriormente. Utilizando a la variable *“verifPasswordLogin”* (declarada en el main e inicializada en 0), si el flag retornado es 1, la contraseña es correcta y el sistema comienza a correr. En caso de ser 0, se mostrará por pantalla *“La contrasenia es incorrecta”*, y volverá a pedirle al usuario que ingrese su contraseña.

Una vez que haya realizado estas tareas de validación, la variable *“seguirFirst”* pasará a valer 1, lo que indicará que se puede continuar con la ejecución.

Por otro lado, si se ingresa la opción 2, se llamará a la función *“cargarArchivoUsuariosConUnUsuario”*, que a su vez llamará a *“cargarVariosUsuarios”*, y esta misma llamará a *“cargarUsuario”*. De esta forma se iniciará el registro de un nuevo usuario.

Al momento de comenzar con el proceso de registración, se le pedirá que ingrese su apellido y nombres. Una vez que el usuario haya ingresado dichos datos, *“cargarUsuario”* llamará a *“verificarNombreUsuario”* que, mediante un flag que será 1 si el usuario ya está registrado o 0 si es nuevo, indicará si se puede continuar con el registro o no.

En caso de que el flag sea 0, se requerirá también de la dirección de e-mail, teléfono celular del usuario y una contraseña de máximo 7 caracteres.

Una vez que el sistema haya recibido todos estos datos, el estado de este nuevo usuario (cuyo ID será determinado por una variable contadora que auto incrementa los ID's de los usuarios mediante el conteo de registros + 1) pasará a ser 1, es decir, activo, y este valor será retornado.

La función *“cargarVariosUsuarios”* recibirá dicho valor y se lo asignará a una variable local *“aux”* de tipo Usuario. Al ser 1, escribirá todos los datos ingresados por el usuario en el archivo.

Por último, *“cargarArchivoUsuariosConUnUsuario”* abrirá el archivo cada vez que los usuarios ingresen nuevos datos, manteniendo estos datos en el mismo.

*“cargarVariosUsuarios”* y *“cargarVariosUsuariosAdmin”* difieren entre sí presentándole al administrador la pregunta *“Desea seguir cargando usuarios?”* por pantalla, manteniendo la ejecución activa mientras la variable controladora sea *“s”*. En cambio, al usuario no se le hace esta pregunta, ya que su proceso de registro será único: no tiene por qué ingresar dos cuentas de una sola vez, como sí podría requerirlo un administrador.

Es por esta misma razón que existen dos funciones *“cargarVariosUsuarios”*, una con un *“Admin”* añadido al final; para darle tanto a usuario como a administrador los privilegios justos y necesarios.

Para validar la existencia de una película en el archivo y evitar que ésta se duplique, existe la función *“verificarNombrePelícula”*, la cual es llamada por *“cargarPelícula”* cuando un administrador ingresa la opción 2 (*“Alta de películas”*) en el apartado “Películas”. El sistema le preguntará si desea dar de alta una película y, en caso de ser una respuesta positiva, le pedirá que ingrese el ID de la película.

*“altaBajaDePelículas”* se encargará de llamar a *“cargarArchivoPelículas”*, que a su vez llamará a *“cargarVariasPelículas”*, función que contiene una llamada a *“cargarPelícula”*. En caso de que la función de verificación dentro de esta retorne un 1, se le advertirá al administrador que se ingresó un nombre repetido, y reiniciará el proceso de carga hasta que se ingrese una película nueva para el sistema. Una vez que se haya ingresado un nuevo título, su año, género, tres actrices o actores, calificación y estado, se retornará el valor 1, que indicará que la película ha sido cargada exitosamente.

*“cargarVariasPelículas”* tendrá una variable *“aux”* de tipo Película que, en caso de ser 1, indicará que se debe escribir una película en el archivo. Se le preguntará al *admin* si desea seguir cargando películas y, mediante la función *“cargarArchivoPelículas”* todas las obras que se ingresen pasarán a guardarse en el archivo.

## **Funciones extra**

### Sonido de inicio de sesión

La función *“sonidoLogin”* utiliza la librería de Windows (incluida por defecto en Codeblocks), tanto para reproducir el sonido como para hacer un “delay” de la ejecución 5 segundos hasta que finalice su reproducción.

### Banner de bienvenida

La función *“imprimirBienvenida”* hace *printfs* línea por línea de un banner que reza “Lab One Prime Video”, y por debajo de este, un cartel que imprime las opciones de “log-in” y “registrarse” para guiar al usuario sobre cuál es el primer paso que debe dar al iniciarse la ejecución del programa.